Lab Exercise 2: Advanced Data Analysis

Athika Fatima - 101502209
Dev Chetal - 101459557
Sreejita Chowdhury - 101590107
Dhruv Jayal - 101503569
Panchasara Mohitkumar Jayeshbhai - 101567404
Shreya Lingwal - 101583877
Tri Thanh Alan Inder Kumar - 101413004
Chirag Vaghasiya - 101505362
Vrunal Patel - 101574257

1. Identify and Document Entities, Attributes, and Referential Integrity

Entities and Attributes:

1. **Orders (Transaction-Level Data)**
    - **Attributes:**
        - `Order ID` (Primary Key)
        - `Order Date`
        - `Ship Date`
        - `Ship Mode`
        - `Customer ID` (Foreign Key to Customer Table)
        - `Product ID` (Foreign Key to Product Table)
        - `Quantity`
        - `Sales`
        - `Discount`
        - `Profit`
2. **Customers**
    - **Attributes:**
        - `Customer ID` (Primary Key)
        - `Customer Name`
        - `Segment`
        - `Region`
3. **Products**
    - **Attributes:**
        - `Product ID` (Primary Key)
        - `Product Name`
        - `Category`
        - `Sub-Category`
4. **Regions (Geographic Data)**
    - **Attributes:**
        - `Region ID` (Primary Key)
        - `Region Name`
        - `City`
        - `State`
        - `Postal Code`

Domains:

- **Order Dates:** Domain constrained to valid dates.
- **Sales, Discounts, Profit:** Numeric domains with ranges (e.g., no negative sales or invalid discounts).
- **Regions:** Only valid regions (e.g., North, West, East) allowed as inputs.
- **Customer Segments:** Limited to pre-defined segments such as **Consumer**, **Corporate**, and **Home Office**.

Referential Integrity (Primary and Foreign Keys):

- **Orders Table:**
  - Primary Key: `Order ID`
  - Foreign Keys:
    - `Customer ID` → Customers Table
    - `Product ID` → Products Table
- **Customers Table:**
  - Primary Key: `Customer ID`
- **Products Table:**
  - Primary Key: `Product ID`
- **Regions Table:**
  - Primary Key: `Region ID`

---

2. Data/Process Flow Diagram (Detailed Description)

Overview:

The Data/Process Flow Diagram (DFD) illustrates how data flows between different processes such as order processing, customer management, and reporting. The following description outlines the main processes and relationships.

Processes:

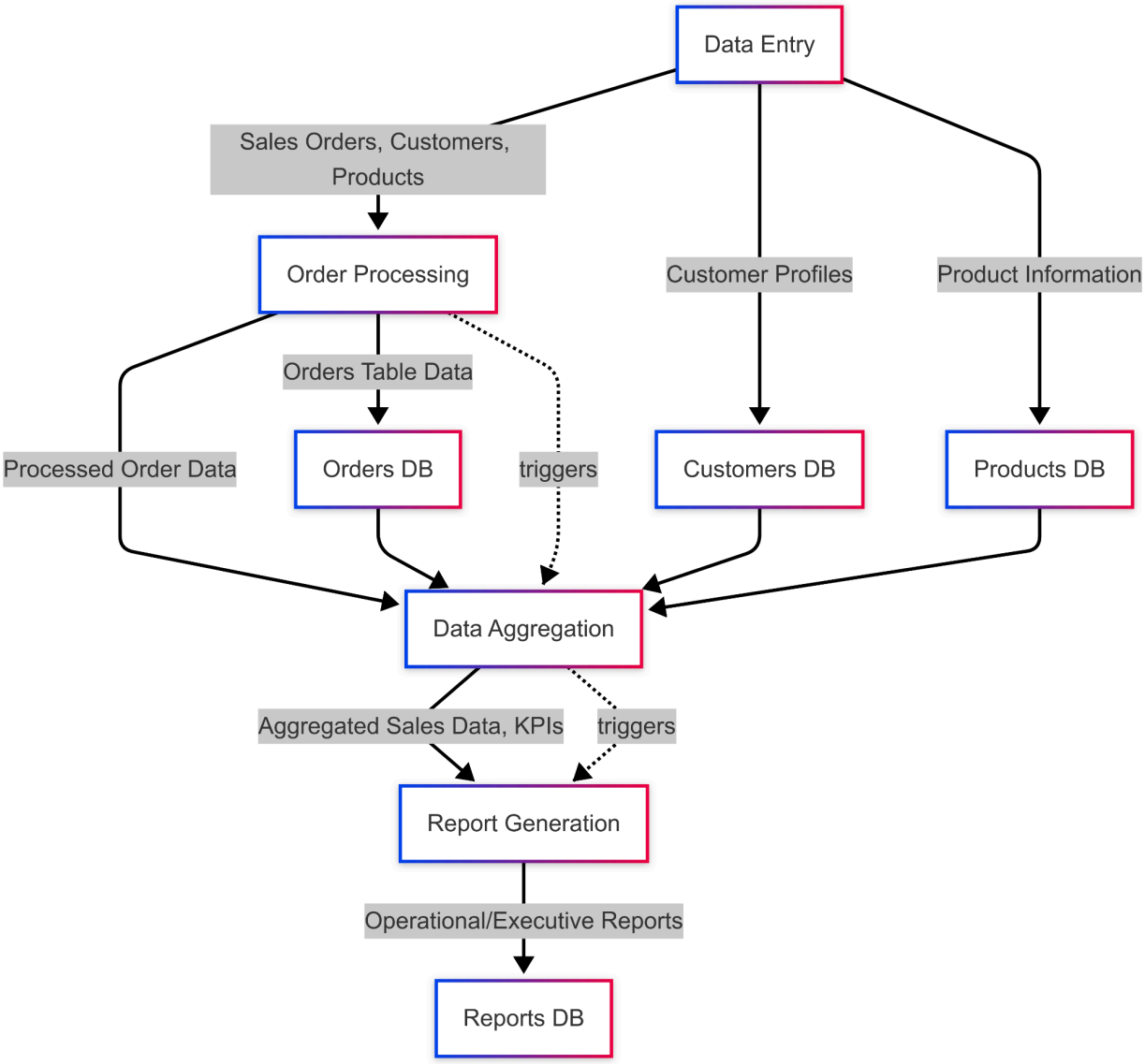
1. **Data Entry Process:**
   - Data is entered through transactions involving sales orders, customer details, and product data.
   - **Input Data:** Sales orders, customer registrations, and product catalogs.
   - **Output Data:** Order confirmations, updated inventory, and performance data.
2. **Order Processing:**
   - **Inputs:** Sales transactions (`Order Date`, `Sales`, `Customer ID`, `Product ID`).
   - **Outputs:** Confirmed orders, updated financial performance metrics.

3. **Data Aggregation for Reports:**
    - **Inputs:** Transaction data (`Orders`), customer data (`Customers`), and product data (`Products`).
    - **Outputs:** Aggregated reports including sales by region, product performance, and profit analysis.
4. **Report Generation:**
    - **Inputs:** Aggregated sales and performance data.
    - **Outputs:** Operational and executive reports for decision-making.

Data Stores:

- **Orders Database:** Stores transaction-level data, including order IDs, dates, and sales amounts.
- **Customer Database:** Stores customer profiles and regions.
- **Product Database:** Stores product details, categories, and sub-categories.
- **Report Data Store:** Stores derived data used for operational and executive reports.

DF Diagram:



---

3. Logical-Level Entity Relationship Diagram (ERD)

Entities and Relationships:

1. **Orders (Transaction-Level Data)**
   - `Order ID` (Primary Key)
   - `Customer ID` (Foreign Key)
   - `Product ID` (Foreign Key)
   - `Order Date`, `Sales`, `Profit`, `Quantity`

Relationships:

- An order is placed by a customer (`Customer ID`), and each order can include multiple products (`Product ID`).

Cardinality:

- One customer can place multiple orders (1-to-many).
- One product can be part of multiple orders (many-to-many, typically resolved with an intersection table if necessary).

2. **Customers**
    - `Customer ID` (Primary Key)
    - `Customer Name`
    - `Segment`
    - `Region`

Relationships:

- A customer is associated with multiple orders through the `Customer ID` key.
- A customer is also linked to a geographic region.

3. **Products**
    - `Product ID` (Primary Key)
    - `Product Name`
    - `Category`
    - `Sub-Category`

Relationships:

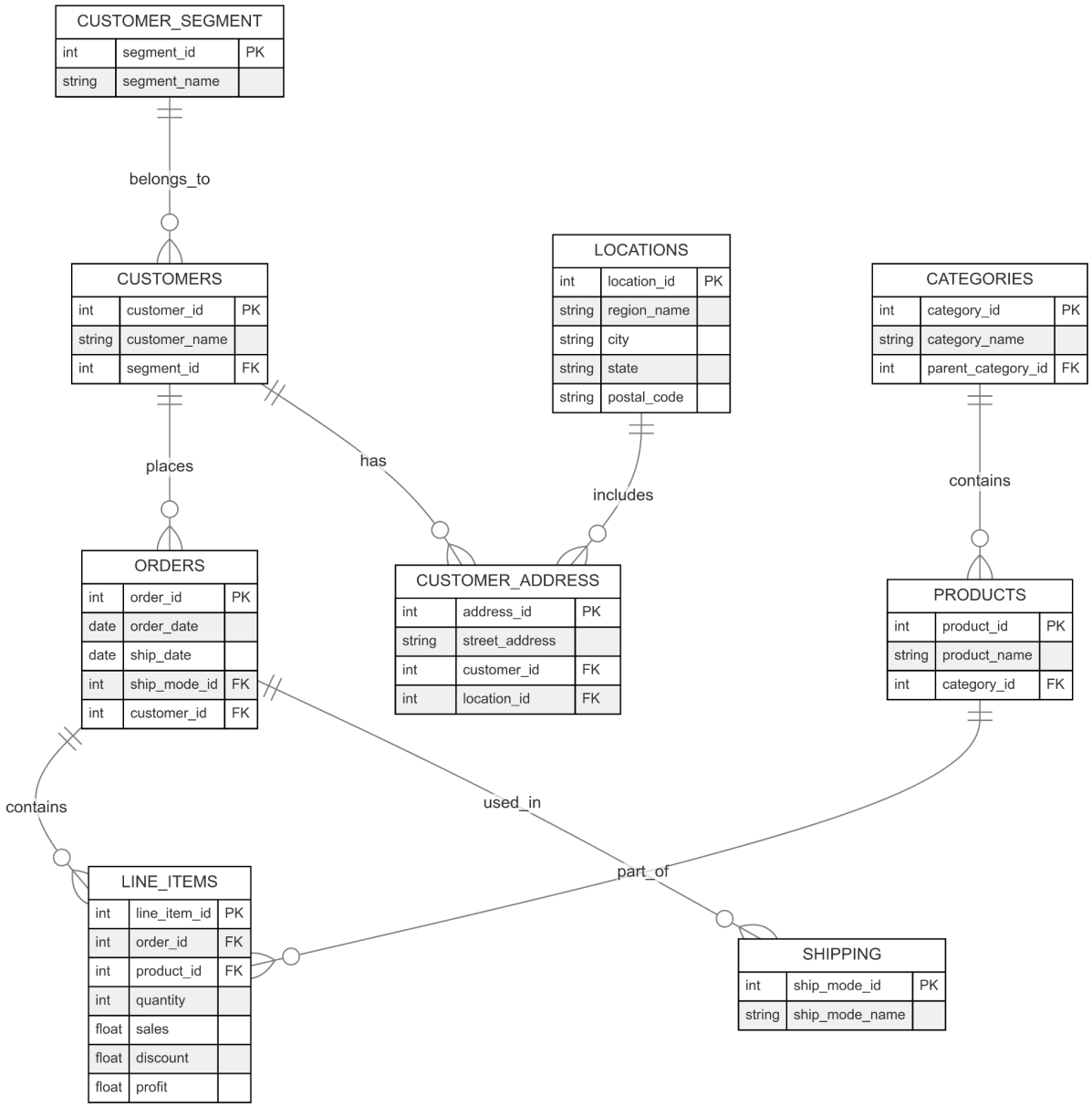
- Products are linked to orders through the `Product ID` foreign key.

4. **Regions**
    - `Region ID` (Primary Key)
    - `Region Name`
    - `City`, `State`, `Postal Code`

Relationships:

- Each region can be linked to multiple customers through the `Region ID` key.

ER Diagram:



---

4. Model Alignment (Physical Table Transformation)

Transformation Steps:

1. **Splitting the `Orders` Table:**

- Split transaction-level data into multiple tables to separate core entities such as `Customers`, `Products`, and `Regions`.
- **Resulting Tables:**
  - `Orders Table`: Retains order-specific data.
  - `Customers Table`: Stores customer information.
  - `Products Table`: Stores product details.
  - `Regions Table`: Stores regional data.

2. **Normalization to 2NF:**
   - Remove partial dependencies by ensuring all non-key attributes are dependent only on the primary key.
   - Example: Move `Region` and `Segment` to their respective tables to avoid redundancy.
3. **Documenting Changes:**
   - Removed redundant columns (e.g., moving region-related columns to the `Regions` table).
   - Created new foreign keys to maintain relationships between split tables.

Change Log Table:

Change Type	Field/Table Affected	Description of Change
New Table	`Regions`	Created a new table to store region-specific information.
Attribute Move	`Region` from `Orders` table	Moved the `Region` attribute to the `Regions` table and linked via `RegionID`.
Attribute Move	`Segment` from `Orders` table	Moved the `Segment` attribute to the `Customers` table to reduce redundancy.
Field Update	Primary/Foreign Keys	Added FKs linking `Orders` to `Customers`, `Products`, and `Regions` tables to maintain referential integrity.
Attribute Deletion	`Region` from `Orders`table	Deleted the redundant `Region` field after moving it to the `Regions` table.
